EEL 5764: Computer Architecture
Project Proposal Group CA Team 1

Project Proposal: Please be clear but concise; it should be a maximum of 2 pages. Submit one proposal file per group, using filename: GroupNameProposal.docx

1. **Names of team members:** Luke Saleh, Nikodem Gazda, Brennan Borchert, Quinlan Stewart
2. **Project title:** HDL Implementation of N-Set Associative Cache System with LRU Replacement
3. **Project description: Type 5: Any computer architecture related project w/ approval from instructor**
4. **Proposed approach** (based on what you know now) Team members expertise and work distribution.
   - **Team Members and expertise:**
     - We will all collaborate when crafting the plan for the design as well as the verification plan.
     - **Nikodem:** Verification. Will handle the individual testbenches for each module.
     - **Quinlan:** Verification. Will handle the integrated testbench for the top-level module.
     - **Brennan:** Design. Will create all the memories/storage, including tag, valid_bit, cache data, and RAM blocks.
     - **Luke:** Design. Will create the LRU buffers and logic as well as the logic for hits/misses.
   - Demonstration plan: based on what you know now.
     - We will be implementing a cache/ram system using HDLs, mainly SystemVerilog and VHDL. In this system, we will have a RAM (of a parameterized size) and an N-set associative cache, configurable at compile time, with the replacement strategy being last recently used. In this design, we will compartmentalize each function by separating each responsibility into separate modules; storage for cache data, cache tags, and cache valid bits will all be separate. The LRU buffers and logic will be contained in one module, but will be separate from the hit/miss logic. These modules will make up the cache block, which will be separate from the RAM.
     - To ensure the functionality of the design, we will create directed and randomized testbenches for each module and then the entire design as a whole. So that our testbenches are powerful and so that we're confident in the design, we will use constrained random verification, not only to focus on edge cases, but also to guide the verification process, for example making sure we read from the same address after we write. We will also use assertions, which will constantly check that the outputs are correct.
   - Proposed schedule for the remainder of the semester*
     - **Stage 1:** Create diagrams/requirements for the design and craft the verification plan.
     - **Stage 2:** Design each module and the testbenches concurrently.
     - **Stage 3:** Conduct testing, debug, and finalize design.
     - **Stage 4:** Document and present.